

A global approach to network and information
security - Investigation / Computer forensics

Pascal Steichen

09/03/2006

Contents

1 Investigation	3
2 Computer forensics	3
2.1 Trust is your enemy	4
2.2 The battle plan	5
2.3 Facing the scene	5
2.4 Prepare for the battle	6
2.5 Documentation and planning	6
2.6 Data capture	7
2.6.1 Volatile data first	7
2.6.2 Non-volatile data	9
2.7 Exploring the abyss	10
2.8 Collect the evidence	11
3 The other side of the medal	12

1 Investigation

- Actions:
 - fight cybercrime
 - perform investigations
- Strategic interest:
 - sovereignty
 - repressives competences
 - international constraints (cybercrime convention, eEurope, i2010, ...)

Actions

- Fight cybercrime. How ?
 - Computer forensics

2 Computer forensics

Flagrante delicto A "flagrante delicto" (caught whilst doing a crime) situation is the ideal for every security office, be it in the physical or in the electronic world.

Often good monitoring, logging, and data capture systems can provide all the info necessary for a catch or at least give a good contextualisation of the scene, for this snort, tripwire, and consorts are your friends, however this isn't implemented much nor very well. Except in dedicated Honeynets.

Computer systems are huge and complex, changing very rapidly and even on well monitored environments things can hide, alarms can be miss leading, etc. The grab to the forensics analysis tools is quasi inevitable.

Definition *Computer forensics is the "art" of gathering and analysing data from a computer crime-scene, in a manner as free from distortion or bias as possible, to determine and reconstruct what has happened on the system.*

The key-factors from this definition, to be kept in mind by every digital-detective, are:

- gathering and analysing

Gathering data is "easy", although one has to be cautious and aware of the implications its actions, understanding how, why and when data is modified.

Analysing is harder and extremely time-consuming.

- crime-scene

This is a more delicate part, cause you can never be sure that there really was a criminal act. Here the legislation and all the legal environment comes in the game and then you really should have a good monitoring system, to proof the break-in (see above).

- free from distortion or bias

Probably the most important part of the definition. The collected data, which later one should or might be used as evidence (in court) has to be as pure as possible.

- reconstruct

Most of the time it is a "post-mortem" issue and the reconstruction of process (the malicious actions) is a key element in understanding what has happened.

2.1 Trust is your enemy



and other lemmas :

- Anything you do to a system disturbs it.
- Keep data as original (unbiased) as possible, but don't work on the original.

- Speed is of the essence - but don't overdo it.
- Be pedantic, but resign yourself to failures.
- Consider the policies, conform to the legal framework.
- Prepare to be surprised.

2.2 The battle plan

To establish the context it's always good to start with questions:

- How has the break-in been performed ?
The modus operandi. Mechanisms, types of attacks like buffer overflow exploitation, back-doors, password cracking, social engineering, etc. the offender used to break-in.
- Why didn't the firewall block the attempt ? Why didn't the IDS send an alarm ?
The door that was left open or the vulnerabilities exploited by the offender to circumvent the security mechanisms. Risks that weren't covered by the security policy in place, ...
- Which are the victim-systems ?
The scope or impact of the attack, machines, persons involved or victimized.
- What data did change on the systems ?
Stolen data, modified data or traces left by the offender.
- What were the rogue's intends ?
Goals and targets of the malicious actions. Motivations of the attacker. This is important to determine the real impacts and legal implications.

2.3 Facing the scene

he analysis of the crime-scene or "situation" must be handled with great care. A fixed and precise procedure has to be followed :

1. Secure and isolate

This first step is somehow converse. Should the network cable be immediately plugged out, or not ? Experts minds are split on this.

2. Record the scene
Respect the order of volatility of data storage.
3. Conduct a systematic search of evidence
Think, define goals, targets (based on logs etc. of unaffected systems). Assume the worst, but move carefully and most important log (write down) every action.
4. Collect and package evidence
Spot and get data that can be used as evidence, by correlating various redundant sources, to be able to draw coherent conclusions.
5. Maintain chain of custody
Keep the parent-child persistence and reconstruct the time line.

2.4 Prepare for the battle

To get a system to an "absolute zero" point (where you could capture all the data in it's original state) is impossible. Think about Schroedinger's cat.

Dan Farmer's prime directive : Strive to capture as accurate a representation of the system(s), as free from distortion and bias as possible.

- If you can't trust the system how can you use the system's tools ?
Even a simple ls could be corrupt) to explore it ? Trust only your tools, or coming from a trustworthy source. For instance statically linked binaries on a CD or other write-protected media. There are some really nice live-CD collections out there, e.g. *Auditor* or *Helix*.
- Off-line or on-line ?
Experts' minds split on this question, maybe a hybrid solution is an answer ?
- First collect, analyse later!
Best is to prepare a secure and trusted machine to collect all the data, for later analysis.

2.5 Documentation and planning

Planning and documenting all the actions during a forensics analysis is paramount, cause sometimes you only have one chance to get an info or respond correctly, especially when working on a hot (on-line) victim.

- A simple table with the following columns can already be very useful:
 - Time
 - Command line
 - Trusted/Untrusted
 - md5sum of data capture
 - Comments

Time	Command line	Trusted	md5	Comments
09/03/2006 10:32:15	' dd < /dev/mem > mem.dd '	Y	689d65e975835...	
09/03/2006 10:38:27	' lsof > of.t '	Y	0c2e968f85600d...	
09/03/2006 10:43:14	' lsof > of.u '	N	4abec5436ca13d...	

- Doing checksums of all transferred data for later consistency is also very important.
- Identical trusted/untrusted execution can be useful to check if the offender did really modify things.

2.6 Data capture

Respect the order of volatility:

1. Registers, peripheral memory,
2. Memory (kernel, physical)
3. Network state
4. Running processes
5. Disk
6. Floppies, backup media, etc.
7. CD-ROMs, printouts, etc.

2.6.1 Volatile data first

- Memory

```
dd < /dev/mem > mem.dd
```

- Kernel messages:

```
dmesg > kmsg.txt
```

- Date and time info of the victim host, to be compared with non-affected hosts

```
date > date.txt
```

- Running processes and open files:

- ps

```
ps aux
```

- lsof

- pcat (included in TCT)

```
pcat 346 | strings > 346.mem  
grep '[pattern]' 346.mem  
strings 346.mem | less
```

- content of the /proc (in /proc/PID/exe the current executables can be found)

- Network config an activity :

- ifconfig

```
eth0 Link encap:Ethernet HWaddr 01:0C:5F:E3:C2:95  
inet addr:192.168.1.18 Bcast:192.168.1.255 Mask:255.255.255.0  
inet6 addr: f480::2dc:6eff:fde3:c295/64 Scope:Link  
UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1  
RX packets:73741 errors:0 dropped:0 overruns:0 frame:0  
TX packets:19848 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:21053599 (20.0 MiB) TX bytes:1738710 (1.6 MiB)  
Base address:0xcf80 Memory:fe9e0000-fea00000
```

- netstat

- route

- w (show who is logged and what they are doing)

- last

```

th      pts/3  vega          Thu Jan  6 09:21 - 17:30 (08:08)
loic    pts/1  asterix.gaule.fr Thu Jan  6 09:14 - 11:26 (02:12)
loix    pts/1  asterix.gaule.fr Thu Jan  6 08:24 - 08:24 (00:00)
ben     pts/1  warp.skynet.be  Thu Jan  6 07:54 - 08:09 (00:14)
andy    pts/1  cpc-ofd2-6-0-c  Thu Jan  6 03:00 - 03:11 (00:10)
bibbi   pts/1  coco.internet.lu Wed Jan  5 22:34 - 22:38 (00:04)

```

– tcpdump (to capture network activity)

- kernel modules/config/patches

– lsmod

– uname

More tips:

- For more automation another TCT tool is very useful : *grave-robber*
- Check if the system is power management enabled, hibernation techniques can save a lot memory to disk.
- A video camera can be useful too.
- At the end of this phase, the network should be switched off, if it hasn't be done yet.

2.6.2 Non-volatile data

A simple backup is not enough, one has to make an identical copy of the filesystem to a secure location. Deleted files are also needed, again dd comes in handy.

Setup up a server on a trusted machine where data should be send to and collected, the netcat tool is ideal for this :

```
[coroner] netcat -l -p 6969 > hda.dd
```

Copying the whole disk to the coroner machine using dd and net(crypt)cat :

```
[victim] dd if=/dev/hda | netcat coroner 6969
```

It is good advice to copy the whole disk not only the partitions (swap is easily forgotten, data may reside on unpartitioned areas). To get partitions separated again (cause many tools only work properly with partitions not disks) :

```
fdisk -lu hda.dd
```

to get start and end cycles

```
dd if=hda.dd of=hda1.dd bs=512 skip=63 count=192717
```

to reconstruct the partition

Another solution can be to use the loopback device-driver, so one doesn't need to split partitions up again.

2.7 Exploring the abyss

Now that the volatile and non-volatile data sources were all securely copied to a trusted host the real analysis can begin. Some examples...

- IDS, firewall and other network log-files analysis:

The network is indeed a very important information source for the coroner, cause the network traces are relatively hard to eradicate (hops via multiple host, which can not all be compromised). However making its way through lines and lines of log-files is not so easy and one important issue is the need to know how system should look like in a normal (uncompromised) state. Further not only log-files are important but all network-related config-files (http.conf, sshd_config, ...) can be very valuable. Useful tools are :

- ethereal
- snort
- tcpreplay, tcpick, tcpdump
- arp (can be useful, think ipspoofing, however arp is easily spoofable too)

It may even be necessary to trace back the network connect through routers, firewalls, to get more logs and detailed info on how and why the break-in did succeed.

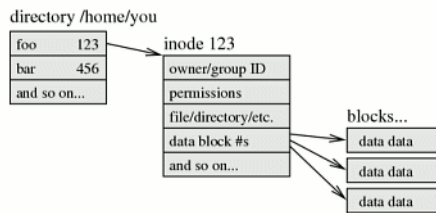
Another very nice information source is DNS, bind for instance keeps track of every query of host. Sending a SIGINT to bind

```
kill -2 [bind-pid]
```

dumps its whole database to file (e.g. named_dump.db).

- Recovering deleted files:

To be able to recover deleted files a deep knowledge of the filesystem is needed, for instance a standard *NIX inode filesystem:



Tools like `ils` and `icat` can find and recover deleted files:

```
ils -f ext -r hda1.dd
```

```
class|host|device|start_time
ils|victim|hda1.dd|1105737210
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_dtime|st_mode|st_nlink
1|a|0|0|1083155632|1083155632|1083155632|0|0|0|0|0|0
30|f|0|0|1075854534|1095519879|1100866281|1100866281|100644|0|548534|2175|2176
31|f|0|0|1083164314|1100866270|1100866282|1100866282|120777|0|18|0|0
32|f|0|0|1075854533|1095519879|1100866282|1100866282|100644|0|933609|2715|2716
33|f|0|0|1083165336|1095519879|1100866282|1100866282|100644|0|149312|3632|3633
34|f|0|0|1083165337|1095519879|1100866282|1100866282|100644|0|512|3779|0
35|f|0|0|1100281361|1100300830|1100866282|1100866282|100644|0|34756|7666|7667
36|f|0|0|1087888466|1100705246|1100866282|1100866282|100644|0|1493636|4710|4711
```

```
icat hda1.dd 35 > 35.dump
```

```
file 35.dump
```

2.8 Collect the evidence

Continuously write down every scrap of unusual happening and try to correlate things together to slowly reconstruct the whole scene and find the modus operandi of the offender. An example of such a scene description is shown in Annex A.

- autopsy

3 The other side of the medal

Who would have thought that IT staff would become the "network cops" ? This simple question nicely defines the scope of the conclusion, because it is indeed not so simple. The legal aspect, especially in the electronic world is becoming more and more important and every of the preceding steps could change the "network cops" to "data spies" or "privacy breakers" etc.

Another issue showing how far computer forensics is still away from the prestige that regular forensics has been granted in our society, is business continuity (e.g. BASEL 2).

Even if computer forensics did evolve nicely in the last years and can in fact reconstruct whole crime-scenes and collect legally valid (hopefully) evidences, keep in mind that developing secure software is at the basis of the "trust chain" contributing far more to the whole IT security system than any attempt to "heal" systems or analyse post-mortem situations.

Annex A

[Example intrusion report](#)